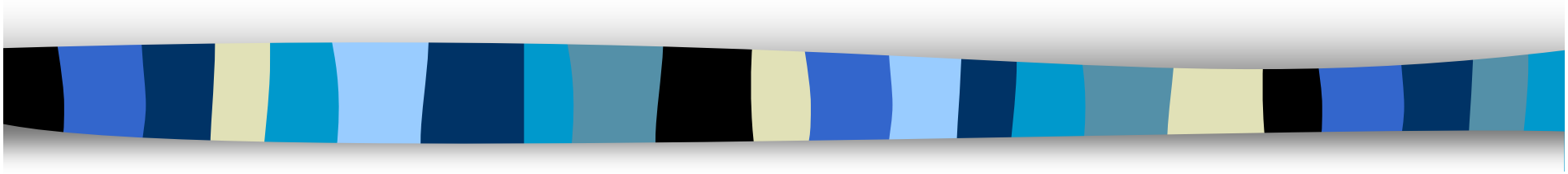
Structured Programming with `goto` Statements



Yann-Gaël Guéhéneuc

Article by Donald E. Knuth

Presented in course IFT6310

Ptidej Team – OO Programs Quality Evaluation and Enhancement using Patterns
Group of Open, Distributed Systems, Experimental Software Engineering
Department of Informatics and Operations Research
University of Montreal

Outline

- Introduction
- Context
- Content
  - Introduction
  - Elimination of `goto` Statements
  - Introduction of `goto` Statements
  - Conclusion
- Discussions
- Map

Introduction

- Article by Donal E. Knuth
  - Published in 1974 in ACM Computing Surveys (since 1969!)
  - 41 pages (!)

```
@article{1241535,
    author = {D. Knuth},
    title = {Structured programming with go to
             statements},
    book = {Classics in software engineering},
    year = {1979},
    isbn = {0-917072-14-6},
    pages = {257--321},
    publisher = {Yourdon Press},
    address = {Upper Saddle River, NJ, USA} }
```

Introduction

- Donald E. Knuth (1938–)
  - Professor Emeritus of the Art of Computer Programming at Stanford University
  - Author of The Art of Computer Programming
  - Creator of TeX and Metafont and “Computer Modern” typeface
  - Proponent of the Literate Programming philosophy of programming
  - …

Outline

- Introduction
- Context
- Content
  - Introduction
  - Elimination of `goto` Statements
  - Introduction of `goto` Statements
  - Conclusion
- Discussions
- Map

Context

- Dawn of structured programming
- Structured programming: **one-in-one-out** rule, where there is only one entrance into a structure, and only one exit out of the structure

- Wonder how it was before?

Context

- In C

```
void copy(char *src, char *dst, int n) {
    while (n > 0) {
        *dst++ = *src++; n--;
    }
}
```

Context

- In C
  - Duff's Device

```
void copy(char *src, char *dst, int n) {
    switch (n & 3) {
    case 0: while (n > 0) {
        *dst++ = *src++;
    case 3: *dst++ = *src++;
    case 2: *dst++ = *src++;
    case 1: *dst++ = *src++;
    n -= 4;
        }
    }
}
```

Context

- In C
  - Duff's Device is legal in C
  - Duff's Device is impossible in Java, which is more structured than C

Context

- In C# (!)

```
using System;

public class SwitchGoto {
  public static void Main() {

    for(int i=1; i < 5; i++) {
      switch(i) {
        case 1:
          Console.WriteLine("In case 1");
          goto case 3;
        case 2:
          Console.WriteLine("In case 2");
          goto case 1;
        case 3:
          Console.WriteLine("In case 3");
          goto default;
        default:
          Console.WriteLine("In default");
          break;
      }

      Console.WriteLine();
    }
  }
}
```

Context

- In Perl, excerpt from the documentation

NAME
SYNOPSIS
DESCRIPTION

NAME

goto - create spaghetti code

SYNOPSIS

goto LABEL
goto EXPR
goto &NAME

Outline

- Introduction
- Context
- Content
  - Introduction
  - Elimination of `goto` Statements
  - Introduction of `goto` Statements
  - Conclusion
- Discussions
- Map

Introduction

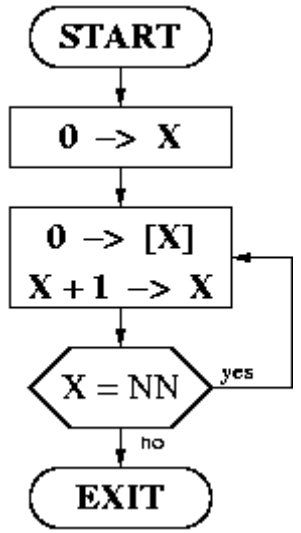
- Revolution of structured programming
  - Main concern: speed!
  - Related to: software crisis (circa. 1960)
  - Fear: restriction of the features in programming languages

Outline

- Introduction
- Context
- Content
  - Introduction
  - Elimination of `goto` Statements
  - Introduction of `goto` Statements
  - Conclusion
- Discussions
- Map

Elimination of `goto` Statements

- Pros
  - Removal of `goto`
    - Replaced by indentation
    - Replacing flow charts
    - Improved maintainability



Elimination of `goto` Statements

- Pros
  - Introduction of `for` loops
    - Concealed loops

```
10 i = 0
20 i = i + 1
30 print i; " squared = "; i * i
40 if i < 10 then goto 20
50 print "Program Completed."
60 end
```

```
for i = 1 to 10
    print i; " squared = "; i * i
next i
print "Program Completed."
end
```

Elimination of `goto` Statements

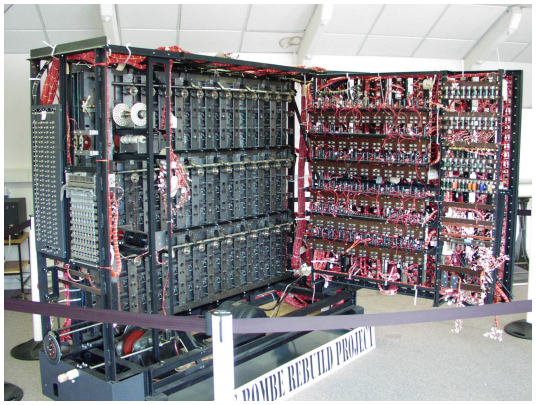
- Pros
  - Improving “quality”
    - “the quality of their programmers was inversely proportional to the density of goto statements in their programs…”
    - Dijkstra’s article “Go to Statement Considered Harmful”

Elimination of `goto` Statements

- Edsger Wybe Dijkstra (1930–2002)
  - Recipient of the 1972 A. M. Turing Award
  - Schlumberger Centennial Chair of Computer Sciences at The University of Texas at Austin (from 1984 to 2000)
  - Fundamental contributions in the area of programming languages
    - Shortest path-algorithm
    - Semaphore construct
    - Self-stabilization

Elimination of `goto` Statements

- ACM A. M. Turing Award
  - Alan M. Turing (1912–1954)
    - Turing Test
    - One of the first designs for a stored-program computer
    - Hut 8 at Bletchley Park (naval cryptanalysis)



Elimination of `goto` Statements

- ACM A. M. Turing

2006 Allen, Frances E

2005 Naur, Peter

2004 Cerf, Vinton G.
Kahn, Robert E.

2003 Kay, Alan

2002 Adleman, Leonard M.
Rivest, Ronald L.
Shamir, Adi

2001 Dahl, Ole-Johan
Nygaard, Kristen

2000 Yao, Andrew Chi-Chih

1999 Brooks, Frederick P.

1998 Gray, Jim

1997 Engelbart, Douglas

1996 Pnueli, Amir

1995 Blum, Manuel

1994 Feigenbaum, Edward
Reddy, Raj

1993 Hartmanis, Juris Stearns, Richard E

1992 Lampson, Butler W.

1991 Milner, Robin

1990 Corbato, Fernando J.

1989 Kahan, William

1988 Sutherland, Ivan

1987 Cocke, John

1986 Hopcroft, John E
Tarjan, Robert E
Tarjan, Robert E

1985 Karp, Richard M.

1984 Wirth, Niklaus E

1983 Ritchie, Dennis M.
Thompson, Kenneth Lane

1982 Cook, Stephen A.

1981 Codd, Edgar F.

1980 Hoare, C. Antony R.

1979 Iverson, Kenneth E.

1978 Floyd, Robert W

1977 Backus, John

1976 Rabin, Michael O.
Scott, Dana S.

1975 Newell, Allen
Simon, Herbert A.

1974 Knuth, Donald E.

1973 Bachman, Charles W.

1972 Dijkstra, E. W.

1971 McCarthy, John

1970 Wilkinson, J. H.

1969 Minsky, Marvin

1968 Hamming, Richard

1967 Wilkes, Maurice V

1966 Perlis, A. J.

Elimination of `goto` Statements

- Cons
  - "We found that there was no way to implement certain simple constructions with while and conditional statements substituted for go to's, **unless extra computation was specified**" [bold mine]

Elimination of `goto` Statements

- Danger
  - Unmaintainable “optimised” code
    - Optimise only the inner loops
    - Allow less efficient but more readable code in non-critical loops
    - “Premature optimisation is the root of all evil”

Elimination of `goto` Statements

- Error exit
  - "Sometimes it is necessary to exit from several levels of control, cutting across code that may even have been written by other programmers"
- Subscript Checking
- Hash Coding
- Text Scanning
- Tree Searching

Elimination of `goto` Statements

- Replacement Proposals
  - Many proposals based on/by Kleene, Jacopini, Ashcroft…
  - “The underlying structure of the program is what counts”
  - Event Indicators
    - `exit`, `jumpout`, `break`, `leave`, `loop…repeat`

Outline

- Introduction
- Context
- Content
  - Introduction
  - Elimination of `goto` Statements
  - Introduction of `goto` Statements
  - Conclusion
- Discussions
- Map

Introduction of `goto` Statements

- Pros
  - Recursion elimination
    - Save space
    - Save time
    - Decrease clarity

Introduction of `goto` Statements

- Pros
  - "Automatic" well-understood transformation
    - Recursion elimination
    - Optimisations that a compiler cannot do
    - Removals of invariant sub-expressions from loops
    - Implementation of "abstract" data structure by "concrete" ones
    - Removals of Boolean variables by code duplication (!)
    - Co-routines

Introduction of `goto` Statements

- Pros
  - Multi-way Branching
    - Micro-programmed emulator
    - Simulator
  - “Fall-through”

Outline

- Introduction
- Context
- Content
  - Introduction
  - Elimination of `goto` Statements
  - Introduction of `goto` Statements
  - Conclusion
- Discussions
- Map

Conclusion

- “The act of focusing our mightiest intellectual resources on the elusive goal of go to-less programs has helped us get our minds off all those really tough and possibly unresolvable problems and issues with which today’s professional programmer would other have to grapple.”

  John Brown [Knuth citation: “In memoriam…”, unpublished note, January 1974]

Conclusion

- "The real issues is structured programming"
  - Correctness proofs
    - Sufficiently convincing
  - Different level of understanding (abstraction)
  - Clarity and understanding
  - "`go to`'s do not have a syntactic structure that the eye can grasp automatically"
  - "Better" language features
  - Efficiency
  - Agreed-upon language by… 1984 (!)